sustainability

MDPI

Article

Hand Gesture Recognition Based on Auto-Landmark Localization and Reweighted Genetic Algorithm for Healthcare Muscle Activities

Hira Ansar [1], Ahmad Jalal [1], Munkhjargal Gochoo [2] and Kibum Kim [3,*]

1 Department of Computer Science, Air University, Islamabad 44000, Pakistan; 191625@students.au.edu.pk (H.A.); ahmadjalal@mail.au.edu.pk (A.J.)
2 Department of Computer Science and Software Engineering, United Arab Emirates University, Al Ain 15551, United Arab Emirates; mgochoo@uaeu.ac.ae
3 Department of Human-Computer Interaction, Hanyang University, Ansan 15588, Korea
* Correspondence: kibum@hanyang.ac.kr

Abstract: Due to the constantly increasing demand for the automatic localization of landmarks in hand gesture recognition, there is a need for a more sustainable, intelligent, and reliable system for hand gesture recognition. The main purpose of this study was to develop an accurate hand gesture recognition system that is capable of error-free auto-landmark localization of any gesture dateable in an RGB image. In this paper, we propose a system based on landmark extraction from RGB images regardless of the environment. The extraction of gestures is performed via two methods, namely, fused and directional image methods. The fused method produced greater extracted gesture recognition accuracy. In the proposed system, hand gesture recognition (HGR) is done via several different methods, namely, (1) HGR via point-based features, which consist of (i) distance features, (ii) angular features, and (iii) geometric features; (2) HGR via full hand features, which are composed of (i) SONG mesh geometry and (ii) active model. To optimize these features, we applied gray wolf optimization. After optimization, a reweighted genetic algorithm was used for classification and gesture recognition. Experimentation was performed on five challenging datasets: Sign Word, Dexter1, Dexter + Object, STB, and NYU. Experimental results proved that auto landmark localization with the proposed feature extraction technique is an efficient approach towards developing a robust HGR system. The classification results of the reweighted genetic algorithm were compared with Artificial Neural Network (ANN) and decision tree. The developed system plays a significant role in healthcare muscle exercise.

Keywords: directional image; geodesic distance; gray wolf optimization; hand gesture recognition; landmark localization; reweighted genetic algorithm; saliency map

check for updates

Citation: Ansar, H.; Jalal, A.; Gochoo, M.; Kim, K. Hand Gesture Recognition Based on Auto-Landmark Localization and Reweighted Genetic Algorithm for Healthcare Muscle Activities. *Sustainability* **2021**, *13*, 2961. https://doi.org/10.3390/su13052961

Academic Editor: Gwanggil Jeon

Received: 10 February 2021
Accepted: 5 March 2021
Published: 9 March 2021

1. Introduction

Recent developments in artificial intelligence and digital technologies have provided several effective ways to communicate in terms of human–computer interaction (HCI). When gestures are made by human body movements, physical actions of fingers, hands, arms, head, and face are recognized by the receiver—this methodology is termed human gesture recognition (HGR) [1–4]. HGR has wide-ranging applications such as communication with and between deaf people, as well as interactions between young children and patients using a PC [5–7]. For rehabilitation purposes, healthcare centers provide hand muscle exercise in which HGR plays a vibrant role. According to the World Health Organization (WHO), 15 million people suffer from stroke and 50,000 people suffer from spinal cord injuries. They affect individuals' upper limb function and also leads to long-term disabilities. Rehabilitation strategy is an essential method for upper limb recovery. HGR is used to perform rehabilitation gestures, and also daily gestures can be recognized [8].

Gestures are extensively characterized as static and dynamic in a natural way of communication [9]. A static gesture is seen at the spurt of time, whereas a dynamic gesture changes with a time frame. The static gestures are specific transition phases in a dynamic gesture that display as specific action or gesture. The gesture can be inferred by a vision-based system and data-glove-based system collected via (i) camera, (ii) sensors, and (iii) gloves [10]. The sensors and gloves measure the angles of the joint and the positions of a finger in real time. The use of gloves and sensors adds a certain burden to the user, and the weight of cables can hinder the movement of the hand, which affects accuracy when measuring gestures. On the other hand, one or more cameras can be used to capture images of gestures performed by an individual. The camera collects static gestures, which are used to train the machine for recognition; for this purpose, only a sufficient dataset is required [11–14].

In this paper, we propose an effective method to extract gestures from RGB images. First of all, preprocessing was performed on all the images. Then, the hand was segmented from the background by two methods, one being a fused method and the other being a directional images method. Both of these methods extracted the hand from the background successfully, but after a comparison of the two methods, the fused method gave better results and was used for further processing. In the second step, landmarks were extracted via color quantization. These landmarks were then used for feature extraction. In this paper, we extracted different features for the accurate recognition of gestures, i.e., angular features, geometric features, and mesh geometry. Those features were optimized and then classified via a genetic algorithm into gestures. The five datasets used for experimentation are named Sign Word, Dexter1, Dexter + Object, STB, and NYU datasets. The proposed system produced significantly better recognition accuracy compared with other state-of-the-art methods.

The main contributions of the paper can be summarized as follows:

- We extracted the hand via a fused method technique from RGB images for gesture classification.
- Auto-landmark localization was performed for multi-feature extraction to improve the feature selection process for daily gestures.
- Multi-features were then optimized via a gray wolf algorithm and classified with a weighted genetic algorithm.
- A comprehensive evaluation was performed on three datasets with significantly better performance than other state-of-the-art methodologies.

The rest of the paper is organized as follows. In Section 2, the literature review is presented on the basis of two main categories of HGR feature extraction and recognition. Section 3 addresses the proposed HGR model, which includes angular, geometric, and mesh geometry-based features; gray wolf optimization; and the genetic algorithm as a classifier. Section 4 discusses the experimental setup and a comparison of the proposed method with other state-of-the-art methods. Finally, Section 5 presents the conclusion and future work.

2. Literature Review

2.1. HGR Through Electromyographic Signals

Human gesture recognition is applied in many research areas because the accurate classification of hand gesture electromyography (EMG) signals provides accurate gesture recognition results [15]. However, the collection of features and the labeling of the large datasets consumes a large amount of processing time. Su et al. [16] proposed a novel method in which they combined depth vision learning and EMG for hand gesture recognition. The system labels data without considering the sequence of hand motion via depth vision learning. The hierarchical k-means (HK-mean) algorithm is used to classify 10 hand gestures using a Myo armband. Motoche et al. [17] used superficial EMG for hand gesture recognition. They applied a sliding window approach; a sub-window is applied to observe signal segments through the main window. The acquired data using Myo armband is then applied to preprocess for rectification and filtering. After that, they extracted features from the feature vector and the results from the functions. They used a feedforward neural network for classification and obtained 90.7% recognition accuracy. Sapienza et al. [18] presented a model with minimum complexity based on the average threshold crossing (ATC) technique. Four movements of the wrist: flexion, extension, abduction, and grasp were detected after the acquisition of signals from EMG. The signal threshold-crossing event number was exploited and then the average ATC classifier produced 92.87% accuracy. Arenas et al. [19] collected data via eight Myo armband sensors with the use of a power spectral density map. For classification, they built a feature set consisting of 2880 multi-channel feature maps, which were divided into three equal sets for training, validation, and testing. Convolutional neural networks (CNNs) obtained 98% accuracy in validation and 99% in testing. Benalcazar et al. [20] identified the labels of hand movements in real time. Their model collected hand movements from a Myo armband, and they used a window-based approach to make feature vectors. For classification, they used *k*-nearest neighbor and a time wrapping algorithm, which achieved 89.5% accuracy. Qi et al. [21] reduced the redundancy of EMG signals and enhanced real-time gesture recognition. They used principal component analysis and General Regression Neural Network (GRNN) for the construction of a gesture recognition system. The authors collected nine static gestures using an electromyographic instrument for the extraction of four kinds of signals. After dimension reduction, accuracy reached to 95.1%

2.2. HGR through Smartphone

The pioneering works of hand gesture recognition through smartphones explored different sensing technologies and feature extraction methods for the improvement of recognition accuracy [22]. Wang et al. [23] used a smartphone as an active sonar sensing system for hand movement recognition. The ultrasonic signal emitted by speakers and the phone's microphone receives an echo that is changed by hand movements. The gesture is identified from the recorded signals. Haseeb et al. [24] introduced a novel machine learning solution for hand gesture recognition. They relied on standard Wi-Fi signals, thresholding filters, and recurrent neural network (RNN); for recognition, the smartphone does not require any change in either the hardware or the operating system. The experimental results included changes in scenarios, as well as network traffic between smartphone and Wi-Fi access points. They classified three gestures with 93% accuracy. Zhang [25] used binary motion gestures methods on a smartphone with an accelerometer. They used only two simple gestures, which were expressed as "0" and "1". They first evaluated four kinds of candidate binary gestures and then they split the accelerometer signal sequence into multiple separate gesture signal segments using signal cutting and a merging algorithm. The segments were then classified using five algorithms, namely, dynamic time wrapping (DTW), naïve Bayes, decision tree, support vector machine (SVM) and bidirectional long short-term memory (BLSTM) networks. Panello et al. [26] addressed the issue faced for gesture segmentation and recognition using a smartphone device. They designed an application that uses low-cost and diffused technologies. They designed a new machine learning algorithm that identifies hand gestures using Hu image moments, invariance rotation, translation, and scaling, all with low computation cost.

2.3. HGR Through Camera

A substantial amount of work has been done on the recognition of static gestures using cameras. For static hand, gesture recognition features are extracted via different methods [27–31]. Features can be extracted using the full hand or by using only the fingers of the hand. This section divides the literature review into two subsections: (i) Section 2.3.1 and (ii) Section 2.3.2.

2.3.1. HGR via Full-Hand Features

HGR of static gestures is a challenging task as the extraction of features from the full hand is a composite and requires a lot of machine training for recognition. Many researchers have presented different methods for gesture recognition of the full hand. Oprisescu et al. [32] proposed a method that extracted the contour of the hand, then calculated convexity and finger positioning from the centroid for gestures. Gesture classification is done via a decision tree on nine different gestures with 93.3% mean accuracy. Yun et al. [33] detected the hand via skin color and angle, combined with Hu invariant moments. For classification, they used a Euclidean distance template-matching technique. Ghosh et al. [34] designed a system in which they segmented the hand in preprocessing. A localized contour sequence (LCS) and block-based features are extracted for better representation of the hand. Those features are combined and an SVM classifier is used for the recognition of static hand gestures. Candrasari [35] extracted the hand via YCbCr values. They extracted features on discrete wavelet transform (DWT) and those features were passed through hidden Markov model (HMM) and k-nearest neighbor (KNN) for classification. Rosalina et al. [36] extracted the hand via contour representation using a glove worn by the user. ANN was applied on the American Sign Language (ASL) and digits from 0–9 for classification. The accuracy rate of gesture recognition was 90%. Lin [37] segmented the hand via a color model, and hand poses were obtained for training purposes. The recognition accuracy was 95.96% for seven hand gestures. Pansare et al. [38] proposed a system that was divided into four stages—preprocessing, hand extraction using the Sobel edge detection method, after which the feature vector is computed via the Euclidean distance between contours. After that, the Euclidean distance is compared with the ground truth and the comparison is done for gesture recognition. Xu et al. [39] proposed a novel hand gesture recognition method in which the hand is extracted via skin-color features, and the arm is removed using distance transformation. Hu moments of the gestures are calculated and SVM is used for classification. This approach produced 95.83% accuracy with eight gestures. Lee et al. [40] introduced a method to extract the hand via wristband-based contour features. A simple feature matching method was proposed to obtain a recognition result. Liu J. et al. [41] proposed a feature-boosting network for estimating 3D hand pose. They used convolutional layers for feature learning; these convolution layers were boosted with a new long short-term dependence-aware (LSTD) module which perceived the dependency on different hand parts. To improve reliability of features representation of each part of hand, the researchers also added a context consistency gate (CCG). They used benchmark datasets to test their system against other state of the art methods.

2.3.2. HGR via Landmarks Features

Many approaches have been proposed to localize hand landmarks as a feature extraction technique for gesture recognition. The majority of existing methods include fingertip detection, which is successfully applied by researchers. Puttapirat et al. [42] proposed a system that extracted important landmarks of the hand in the image. They identified the location to specify those landmarks, and the landmarks were matched with the corresponding landmarks in a 3D model to estimate the hand posture. Ma et al. [43] designed a method that extracted region of interest (ROI) by the local neighbor method. They used the convex hull detection algorithm for the identification of fingertips. Al Marouf et al. [44] developed a novel approach to determine the fingertips and the center of the palm. The procedure of fingertip detection is performed via an adaptive hill-climbing algorithm applied on distance graphs. Finger identification is performed via the relative distances between fingers and valley points. Mahdikhanlou et al. [45] explained a novel multimodal framework that computed two sets of features. The first set of features is angles from the hand joints and the second set of features is from hand contours. Those features are then classified using random forest. Grzejszczak [46] proposed a method for the localization of landmarks in RGB images. They analyzed a skin-masked directional image using hand transform and template matching. They detected landmarks on both contour and inside of the hand masks. Recognition is done by computing the localization error of the landmark. Kerdvibulvech [47] made a tracking system for fingertips. They achieved detection by matching a semicircular template to the detected skin region while for classification they used Bayesian classifiers. Nguyen et al. [48] made a system to segment the hand using color information separated from the arm. Then, features were extracted, namely, ratio of width to height, wrist angle, and the number of fingers; calculations are based on fingertips and cross-sections. SVM was applied for classification and they achieved 89.5% accuracy.

3. Materials and Methods

The proposed system is comprehensively discussed in this section. The system is divided into various phases. The HGR system starts with the preprocessing phase, where the hand gesture from each RGB image is segmented from the background using a morphological operation. A fused method is used for hand detection. Next is the feature extraction phase, where geodesic distance, landmarks, geometric features, and spatial features are extracted from processed RGB images. Then, the optimization phase results in a representation of features in the vectorized form via a gray wolf optimization algorithm. Finally, in the classification phase, each gesture is classified via a reweighted genetic algorithm. The overall architecture of the HGR system is shown in Figure 1.

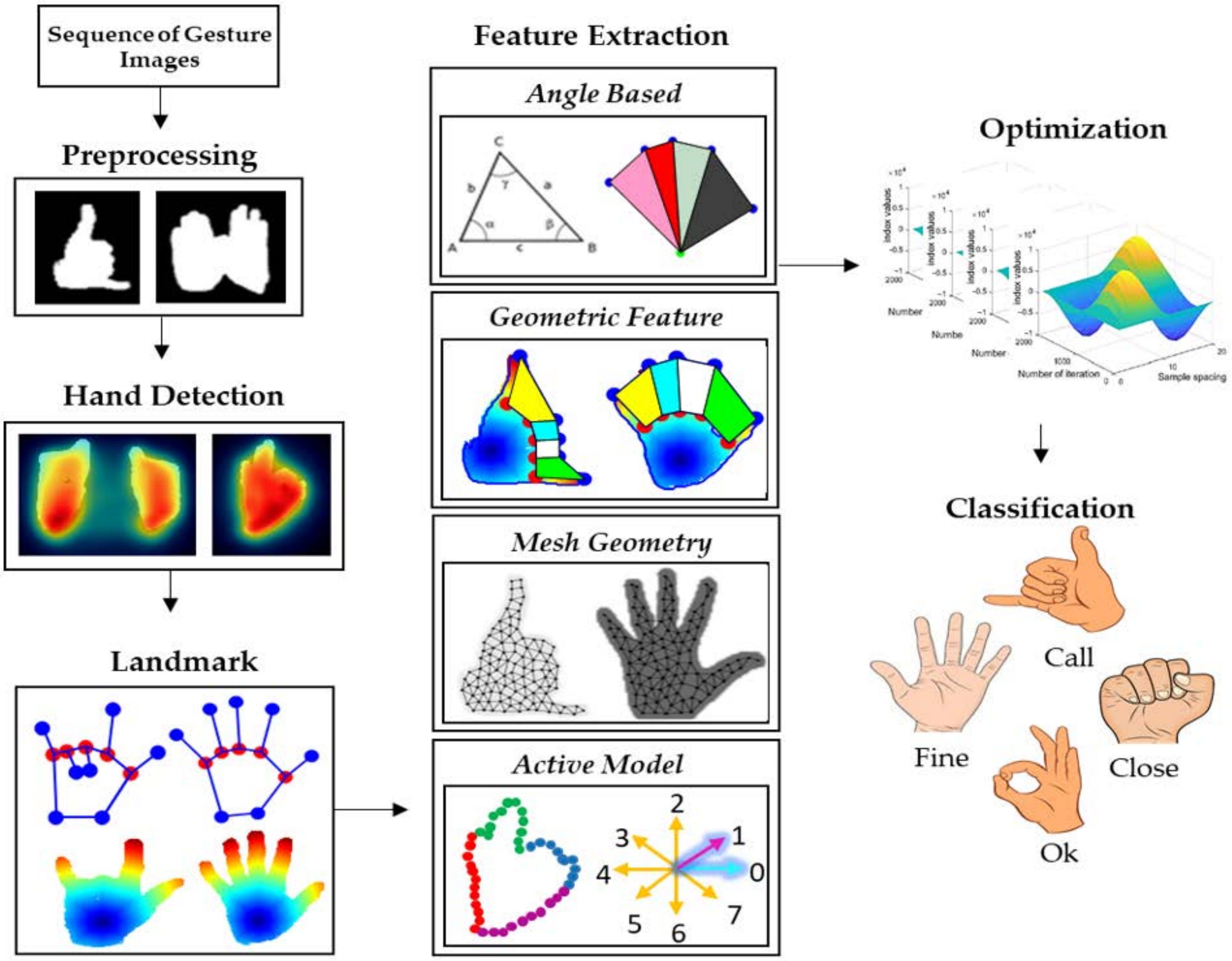


Figure 1. Flow chart of the proposed hand gesture recognition (HGR) system.

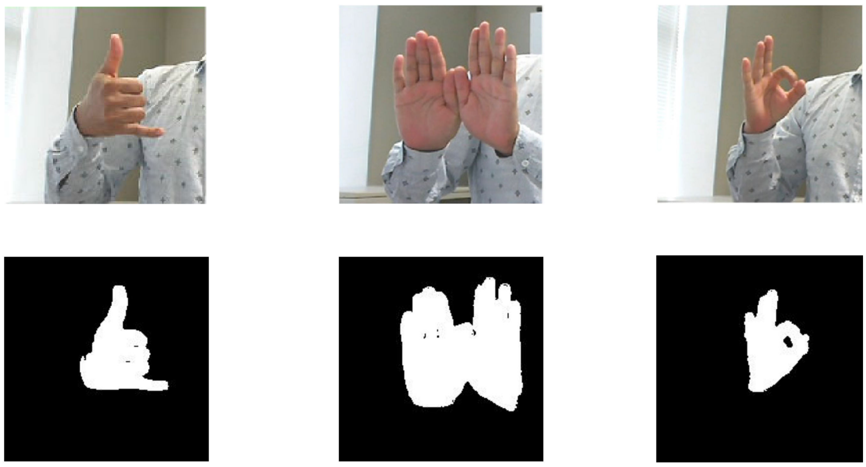
3.1. Preprocessing

RGB images are prone to having noise. This makes the extraction of a region of interest from the background a challenging task. We can extract the ROI by preprocessing, in which first of all noise is removed from the image. Then, a sharpening and enhancement technique is used to increase the intensity and brightness of the image. This image is then converted into binary form for further processing in designed the HGR system. In this phase, a connected component is applied to select the largest component in the image. Then, morphological operations, namely, dilation and erosion, are used to extract the desired region of interest [49].

$$X \oplus Y = \{z|(\hat{Y})_z \cap X \neq \varphi\} \tag{1}$$

$$X - Y = \{z|(Y)_z \subseteq X\} \tag{2}$$

where Y is the structuring element and z is the location of the set of pixels. During the translation of z, the reflection $\hat{Y}$ of Y joins with the pixels of the foreground element X. In this phase, the shape of the object is maintained, and the gesture mask is extracted. Images of all 5 datasets are passed through this phase that has enhanced the images at the pixel level for further processing. Preprocessing results are shown in Figure 2.



(**a**) (**b**) (**c**)

Figure 2. Enhanced and binary images of three gestures in sign word dataset: (**a**) call, (**b**) close, and (**c**) correct.

3.2. Hand Detection

Region of interest (ROI) extraction is the first step in any HGR system [50]. Thus, the ROI, either single or both hand gestures in all RGB images, is first extracted from the background using 2 methods. The 2 methods implemented to segment gesture silhouettes are separately described in the following subsection.

3.2.1. The Fused Method

RGB silhouette extraction of all 5 datasets is carried out through the fused method for hand detection. This method involves 2 methods of detection. Firstly, the entire image dimension is reduced to two-dimensional space where the column size is defined by width and rows are defined by heights in an image. The RGB image is then divided into planes and converted into YCbCr space where the color of each pixel is 32 bits. For the extraction of each channel, right shift is performed on red, blue, green, and alpha by 24-bit processing to obtain the values of alpha. The alpha channel is used to check the opacity of the image; if the pixel has 0% value, then it is fully transparent, whereas if it has 100% values, then it is a fully opaque pixel. For the red and green channel, 16 bits and 8 bits right shift is performed, respectively. The remaining pixel values are for the blue channel.

On these calculated values, bitwise logical AND operation with 0 xff is applied to extract the desired color. These operations are applied to all image pixels [51]. To obtain accurate and more precise recognition, we converted the *IRGB* image into YCbCr color space as in the equation given below:

$$\begin{bmatrix} Y \\ Cb \\ Cr \end{bmatrix} = \begin{bmatrix} 16 \\ 128 \\ 128 \end{bmatrix} + \frac{1}{265}\begin{bmatrix} 65.738 & 129.057 & 25.06 \\ -37.945 & -74.494 & 112.43 \\ 112.439 & -94.154 & -18.28 \end{bmatrix}\begin{bmatrix} R \\ G \\ B \end{bmatrix} \tag{3}$$

where Y is the luminance. To overcome the interference of highlights, Y ϵ (0, 80) is set. Then, using an elliptical equation, human skin color is located via Cb and Cr values. The equation is shown below:

$$\begin{cases} \frac{(i-1.6)^2}{26.39^2} + \frac{(j-2.41)^2}{14.03^2} < 1 \\ \begin{bmatrix} i \\ j \end{bmatrix} = \begin{bmatrix} \cos(2.53) & \sin(2.53) \\ -\sin(2.53) & \cos(2.53) \end{bmatrix}\begin{bmatrix} Cb-109.38 \\ Cr-152.02 \end{bmatrix} \end{cases} \tag{4}$$

where i and j are the intermediate values. Each pixel value of IRGB and YCbCr is compared with the standard skin pixel, and a decision, whether each pixel is skin or not, is made on the range of predefined threshold value for each parameter.

Secondly, a contrast-based method is applied to compute a saliency map. In a saliency map, the dominant part of the gesture is based on saliency values, making the segmentation of gestures faster and more accurate. The algorithm designed for computing the saliency map has 3 aspects: (1) contrast will depend on the color and the area of the two partitions in the image; (2) the partitions will have a greater impact on each other's saliency if the distance between them is closer; (3) the proximity of the saliency object to the center of the image [52]. Saliency map computation is performed by segmenting the input image into super pixels. Then, a sparse color histogram of the super pixels is constructed and the color number of each channel is reduced to simplify the calculations. Each histogram is converted into lab space, and the differences of color and distance between pixels are then calculated.

$$Dis_{pixel}(p_i, p_j) = \sum_{x=1}^{k_1}\sum_{y=1}^{k_2} f(d_{i,x}) f(d_{j,y}) Dis(d_{i,x}, d_{j,y}) \tag{5}$$

where $Dis(d_{i,x},\ d_{j,y})$ is the distance between color x and y in super pixel p_i and p_j. K_1 and K_2 represent the color number of the super pixels p_i and p_j.

As the partitions have a greater impact on each other's saliency map, thus the distance between p_i and p_j is computed as

$$Dis_d(p_i, p_j) = |m_i - m_j| + |n_i - n_j| \tag{6}$$

$Dis_d(p_i, p_j)$ is the distance between regions. m and n represent the X and Y coordinate values of region p_i, respectively.

$$SuperP(p_i) = \sum_{t \neq i} \frac{n(p_t) Dis_i(p_i, p_t)}{\delta Dis_d(p_i, p_t)} \tag{7}$$

where $n(p_t)$ is the total number of super pixels p_t. The greater value will represent the greater impact on each super pixel. The original image is then segmented using graph-based segmentation to obtain a larger partition, and the contour of the salient object is generated from the saliency map [53]. Then, the gray values of the saliency map are merged in the contour. The resultant saliency map is then represented on the top of the original image, as shown in Figure 3.

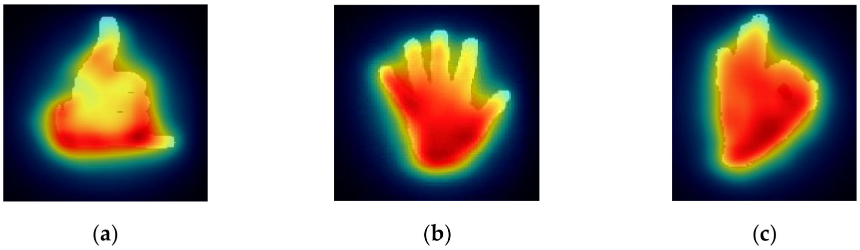


Figure 3. Fused methods of three gestures in sign word dataset: (**a**) call, (**b**) fine, and (**c**) correct.

3.2.2. Directional Images

In the second hand-detection method, the outer and inner edges of the hand region are detected via **a** new approach. The ROI is obtained by specifying a threshold value T, which compares foreground and background pixel values, and as a result, a binary image is generated. For hand detection, a 3*3 gradient vector-matrix $G_{(x,y)}$ is computed. The gradient vector matrix is computed for every pixel of image I, and is represented as follows:

$$G_{(x,\,y)} = \begin{bmatrix} I_{(x,\,y)} - I_{(x,\,y-1)}\,,\; I_{(x,\,y)} - I_{(x,\,y+1)} & I_{(x,\,y)} - I_{(x-1,\,y)}\,,\; I_{(x,\,y)} - I_{(x+1,\,y)} \\ I_{(x,\,y)} - I_{(x-1,\,y-1)}\,,\; I_{(x,\,y)} - I_{(x+1,\,y+1)} & I_{(x,\,y)} - I_{(x-1,\,y+1)}\,,\; I_{(x,\,y)} - I_{(x+1,\,y-1)} \end{bmatrix} \quad (8)$$

Every second pixel in the second row of the matrix will be compared with the distances adjacent to the pixel in the 3×3 window, resulting from the gradient vector matrix. The negative values of distances of every pixel, calculated after subtraction, will be converted into positive.

$$dl = dl \, * \, (-1) \quad (9)$$

The distances that lie vertically, horizontally, and diagonally in the gradient vector are compared with a constant threshold. A distance greater than the threshold is set to the white pixel value of 1, and distances less than the threshold are set to the black pixel value 0 and, as a result, a binary image is formed of outer and internal boundaries of the hand [54]. The resulting directional image is shown in Figure 4.

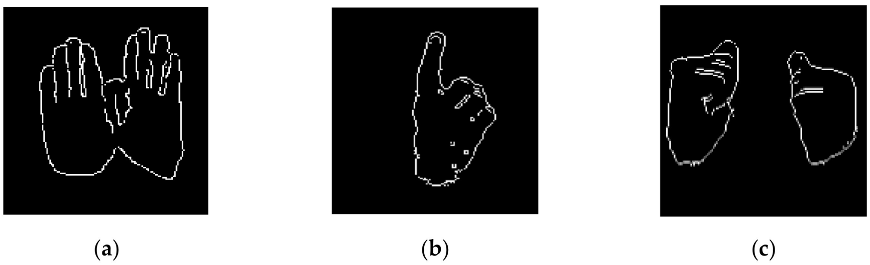


Figure 4. The directional image of gestures of the Sign Word dataset: (**a**) close, (**b**) single, and (**c**) cold.

Both methods were tested on the Sign Word, Dexter1, Dexter + Object, STB, and NYU datasets. The fused method gave more promising hand detection results than the directional image method. The ground truth of gestures is first computed in order to compute the accuracy of the resultant hand detection images for both fused and directional image methods. Then, the contour pixel index values distance is compared via geodesic distance on both of the methods. Table 1 shows comparisons of detection accuracy for the Sign Word dataset. It is clearly shown that the fused method produced more accurate results. Thus, the fused method was selected for further processing of the system architecture.

Table 1. Comparison of detection accuracy for the Sign Word dataset.

Classes	Fused Method Accuracy (%)	Directional Image Accuracy (%)	Classes	Fused Method Accuracy (%)	Directional Image Accuracy (%)
call	92	69	**ok**	92	68
close	91	62	**please**	93	58
cold	92	67	**single**	92	63
correct	92	60	**sit**	92.5	52
fine	92	66	**tall**	91	56
help	92	59	**wash**	92	62
home	91	55	**work**	91	61
like	92	65	**yes**	91	60
love	91.5	65	**you**	92	62
no	91	66	**iLoveYou**	92	68

3.3. Landmark Detection

The segmented hand is then used for landmark detection. Many approaches are proposed to localize hand landmarks, which help in feature extraction for distinguishing and determining specific gestures [54–58]. The majority of techniques are quite simple and limit the exact localization of landmarks. In our proposed method, landmark detection is performed using 2 different methods on different segmented images for the more exact localization of landmarks.

Geodesic Distance

In this method, gestures performed by hands are represented via geodesic wave maps. These maps are generated by calculating geodesic distance found by a fast-marching algorithm. First of all, the center points of a human hand silhouette are located, and the distance value is given as $d\ (h) = 0$. Point h is the starting point, which is marked as a visited point. All the other pixel points p are unvisited and given a distance value $d\ (p) = \infty$ on hand silhouettes. The neighbor of each pixel p is represented as n, and p pixel distance is measured from n. Every neighboring pixel is taken in each iteration until all pixel points are marked "visited" [55–59]. The distance calculated from each iteration is compared with the distance of previous iterations. Priority is given to the shortest distance calculated. An updated distance is defined as

$$d = \begin{cases} \dfrac{d_x + d_y + \sqrt{\Delta}}{2} & when \quad \Delta \geq 0 \\ min(d_x, d_y) + w \ \ otherwise \end{cases} \tag{10}$$

$$\Delta = 2w^2 - (d_x - d_y)^2$$

where d_x and d_y is the distance in x- and y-coordinates, respectively, $d_x = min(D_{i+1,m}, D_{i-1,m})$ and $d_y = min(D_{i,n+1}, D_{i,n-1})$. Figure 5 demonstrates the wave propagation of geodesic distance via fast-marching algorithm (FMA).

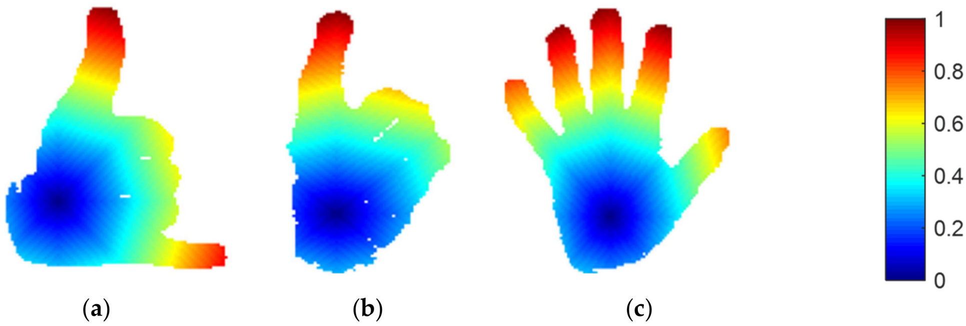


Figure 5. Wave propagation of geodesic distance via fast-marching algorithm (FMA) on the Sign Word dataset classes of (**a**) call, (**b**) single, and (**c**) fine.

Landmark detection is performed after obtaining the wave propagation of geodesic distance via the fast-marching algorithm (FMA) on images. Color values of pixels p are computed on the outer boundary b of the hand silhouettes. Pixels having same color values c are counted first and then the mean is computed; on the mean value of the pixel, the landmark l is drawn. For the inner landmark, the color value of neon green is taken, and the distance is set between points. The fingertips can be calculated as

$$l = \frac{\mathrm{c}\left(\mathrm{p_x}, \mathrm{p_y}\right)}{2} \tag{11}$$

where $\mathrm{p_x}$ and $\mathrm{p_y}$ belong to the same color in the outer boundary and $\mathrm{c}\left(\mathrm{p_x}, \mathrm{p_y}\right)$ is the total number of that colored pixel located in the outer boundary. Landmarks are drawn on the hand silhouettes in Figure 6 given below:

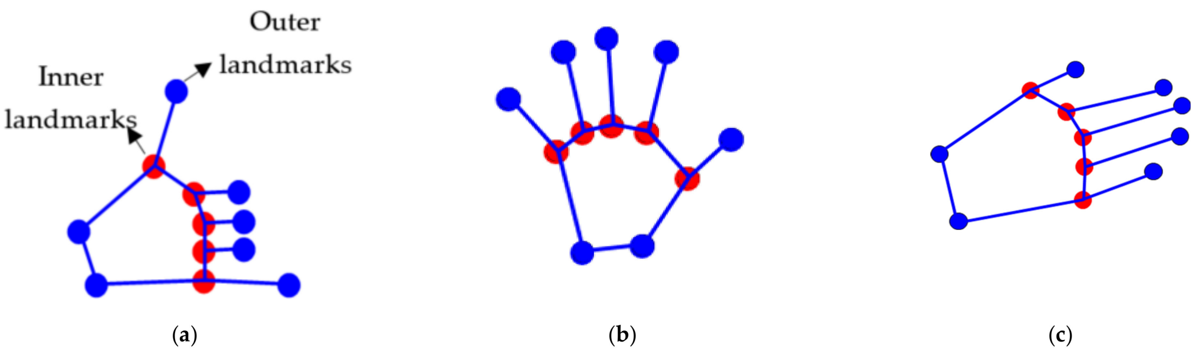


Figure 6. Landmarks detection on the Sign Word dataset with (**a**) call, (**b**) fine, and (**c**) please hand gestures.

3.4. Feature Extraction via Point-Based Method

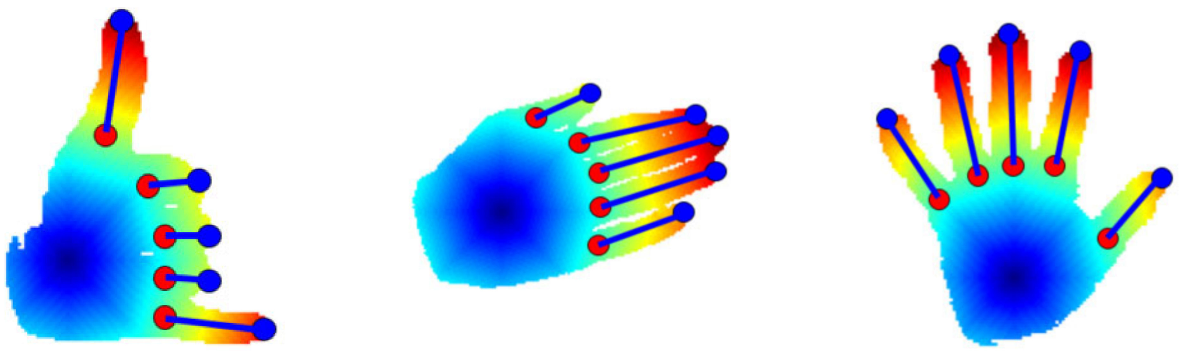
This section provides a detailed description of feature extraction via landmarks. The landmarks are extracted by a point-based features extraction method for hand gesture representation, training, and recognition.

3.4.1. Distance Features

Feature extraction for hand gestures is achieved via the point-based method, which includes points on the thumb, index finger, middle finger, ring finger, and little finger (see Figure 7). All the points are combined in various ways to produce a variety of features that are extracted for training and recognition purpose. These points are distance features, geometric features, and angle-based features. The distance feature d measures the distance between the *ixy* extreme landmark on the fingertip and the *cxy* inner landmark, using that geodesic distance of the hand, which is formulated as

$$\parallel d \parallel = \sqrt{(xi_2 - xc_1)^2 + (yi_2 - yc_1)^2} \tag{12}$$

where d represents the distance between two points; xi_2 and xc_1 are the x-coordinates of the extreme landmark and inner landmark of the hand, respectively [60,61]; while yc_1 and yi_2 are the y-coordinates of the same landmarks.

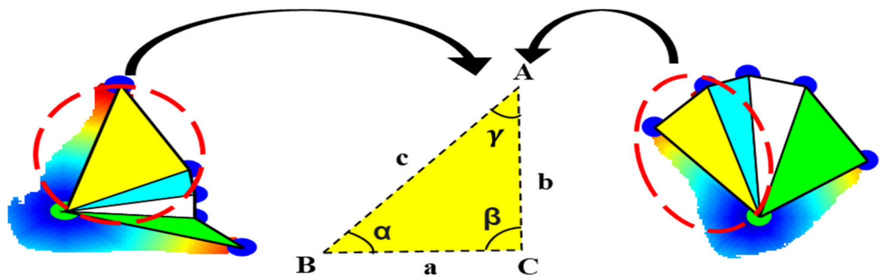


(**a**) (**b**) (**c**)

Figure 7. Distance feature computed on gestures of the Sign Word dataset classes of (**a**) call, (**b**) please, and (**c**) fine.

3.4.2. Angular Features

The angular features are extracted through the cosine of the angles (i.e., α, β, γ) that is measured on the geodesic distance angle of 2 extreme points [62–64]. Three points —adjacent, side, and centroid—form a triangle, as shown in Figure 8b.



(**a**) (**b**) (**c**)

Figure 8. Angular features extraction from triangles drawn on two classes of the Sign Word dataset: (**a**) call, (**b**) angle description, and (**c**) fine.

Similarly, we have vertices, i.e., A, B, and C, and a, b, and c are the sides of a particular triangle, as shown in Figure 8a, having $a = \overline{BC}$, $b = \overline{AC}$, and $c = \overline{AB}$, respectively [65].

$$\alpha = \cos^{-1} b^2 + c^2 - a^2/2bc\beta = \cos^{-1} a^2 + c^2 - b^2/2ac\gamma = \cos^{-1} a^2 + b^2 - c^2/2ab \tag{13}$$

where α, β, and γ are the measures of the angle between two adjacent sides b<->c, a<->c, and a<->b of the triangle formed, respectively. Finally, these features are provided to the classifier for further processing towards recognition, which is discoursed successively [66].

3.4.3. Geometric Features

Hand gestures are formed using different combinations of fingers and palms, which result in forming different shapes. These shapes form a specific geometry over different gestures. Such geometric shapes are the best features for the classification and recognition of gestures [67–70]. The geometric feature is the third point-based feature that includes different irregular shapes formed by 2 consecutive fingers. This includes different irregular shapes formed by 2 consecutive fingers of the hand in a gesture. The area is computed on the shape formed via Heron's formula.

In this method, the irregular shape is simply divided into regular shapes such as a polygon, which is divided into 2 triangles [71]. Each side distance of the triangle is measured as the distance calculated between 2 points, and the values are computed with Heron's formula:

$$G = \sqrt{t\,(t-m)(t-n)(t-o)} \quad where\ t = \frac{m+n+o}{2} \tag{14}$$

where m, n, and o are the sides of the triangle, as shown in Figure 9. After the area of each triangle is calculated, the areas of both triangles are added together to find the area of the irregular shape. In this way, all the shape areas of the various shapes are computed, and the features are then available for classification and recognition.

3.5. Feature Extraction via Full Hand

This section provides a detailed description of feature extraction from the full hand using the index values of points drawn using self-organizing neural gas.

3.5.1. Mesh Geometry

The aim of this stage is to estimate the morphology of the hand. This is accomplished by applying self-organizing neural gas (SONG) on the segmented binary image. SONG is an unsupervised learning model used in applications in which it is important to maintain the topology between input and output spaces. The clustering of input data is achieved so that the distance of the data item in inter-cluster variance is small, and in different classes, inter-cluster variance is large [72–74].

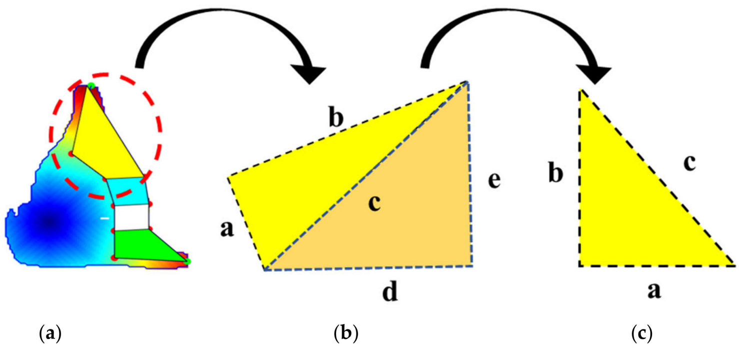


Figure 9. (**a**) Geometric feature collected from irregular shapes of the Sign Word dataset class call; (**b**) irregular shape divided into different sized triangles; (**c**) single triangle side representation.

A typical SONG training starts with the first two output neurons (n = 2). For training of the SONG, all the training datasets I are circularly used. All accumulated errors $E_w^{(1)}$, $E_w^{(2)}$, $\forall\, w\ \varepsilon\ [1,\ n]$ are set to zero from the beginning of each epoch. $E_w^{(1)}$ shows the total quantization error that corresponds to the neuron at the end of an epoch, while the increment of the total quantization error we obtain after removal of the neuron is represented by variable $E_w^{(2)}$. For the given input vector I_x, the starting two neurons are obtained by

$$Neuron_{a1} = \|\ I_x - W_{a1}\ \| \leq \|\ I_x - W_w\ \|,\ \forall\, w\ \varepsilon\ [1,\ n] \tag{15}$$

$$Neuron_{a2} = \|\ I_x - W_{a2}\ \| \leq \|\ I_x - W_w\ \|,\ \forall\, w\ \varepsilon\ [1,\ n]\ and\ w \neq y1 \tag{16}$$

where the initial weight vector W_w, w = 1, 2 are randomly selected by the two neurons in the input space. The values of the local variables $E_{a1}{}^{(1)}$ and $E_{a1}{}^{(2)}$ change according to the following equation:

$$E_{a1}^{(1)} = E_{a1}^{(1)} - \parallel I_x - W_{a1} \parallel E_{a1}^{(2)} = E_{a1}^{(2)} - \parallel I_x - W_{a2} \parallel C_{a1} = C_{a1} + 1 \quad (17)$$

The counter C_a is assigned a zero value for these two neurons, as w = 1, 2, and if $C_{a1} \leq C_{idle}$, then the local learning rate is defined as

$$\varepsilon 2_{a1} = \frac{\varepsilon 1_{a1}}{r_{a1}} \quad (18)$$

where $\varepsilon 2_{a1}$ and $\varepsilon 1_{a1}$ change the value according to (17), (18), and (19). Otherwise, the local values will have constant values $\varepsilon 1_{a1} = \varepsilon 1_{min}$ and $\varepsilon 2_{a1} = 0$.

$$\varepsilon 1_{a1} = \varepsilon 1_{max} + \varepsilon 1_{min} - \varepsilon 1_{min} \cdot \left[\frac{\varepsilon 1_{max}}{\varepsilon 1_{min}}\right]^{\frac{I_{a1}}{I_{idle}}} \quad (19)$$

$$r_{a1} = r_{max} + 1 - r_{max} \cdot \left[\frac{1}{r_{max}}\right]^{\frac{I_{a1}}{I_{idle}}} \quad (20)$$

The learning rate $\varepsilon 1_w$ is applied to the winner neuron, while $\varepsilon 2_w$ is applied to the weights of the neighbor of the winning neuron. The learning rate changes values from maximum to minimum, which is defined by the I_{idle} parameter. The initial value of r_{min} = 1, with the period of time the value of r_{aw} defined by I_{idle} parameter, reaches to maximum r_{max}. The weight vectors of the winning neuron $Neuron_{a1}$ and its neighbor neurons $Neuron_o$, o ϵ ne(a1) are adapted according to the following equations:

$$W_{a1} = W_{a1} + \varepsilon 1_{a1} \cdot (I_x - W_{a1}) \quad (21)$$

$$W_o = W_o + \varepsilon 2_o \cdot (I_x - W_o), \ \forall\, o \in ne(a1) \quad (22)$$

After neurons $Neuron_{a1}$ and $Neuron_{a2}$ are detected, the connection between them is created. At the end of each epoch, all the neurons are in the idle state. If the local counters are greater than the value of C_{idle} then the neurons are well trained. Here, the convergence SONG network is assumed. Figure 10 shows the topological features of input space I extracted by SONG.

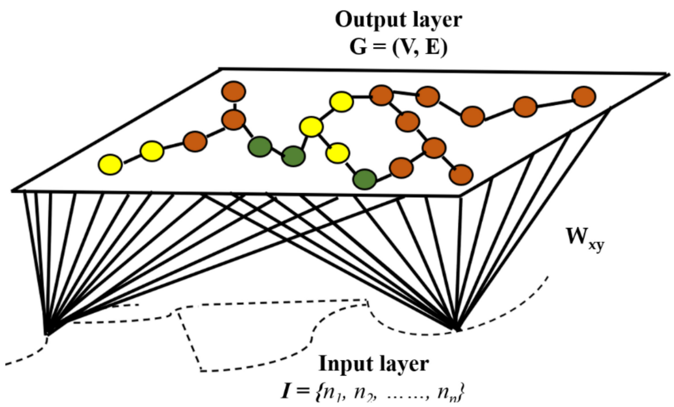


Figure 10. Self-organizing neural gas (SONG) extracted on the input space.

Algorithm 1. Pseudo code for self-organizing neural gas

Input: Input space, *I;*
Output: the map, G = (V, E);
Initialization:
First, randomly generate two nodes, N = (n_1, n_2) in the input space.
Second, set the neighboring neuron to zero and set the maximum number of nodes to 100.
1. Randomly generate one input signal Ÿto update input space I, calculate the winning node x_1 and x_2 nearest to Ÿ

$$x_1 = argmin_{n\, \in N}\ \parallel\ U -\ w_n\ \parallel$$
$$x_2 = argmin_{n\, \in \{x_1\}}\ \parallel\ U -\ w_n\ \parallel$$

2. Adjust x_1 and x_2

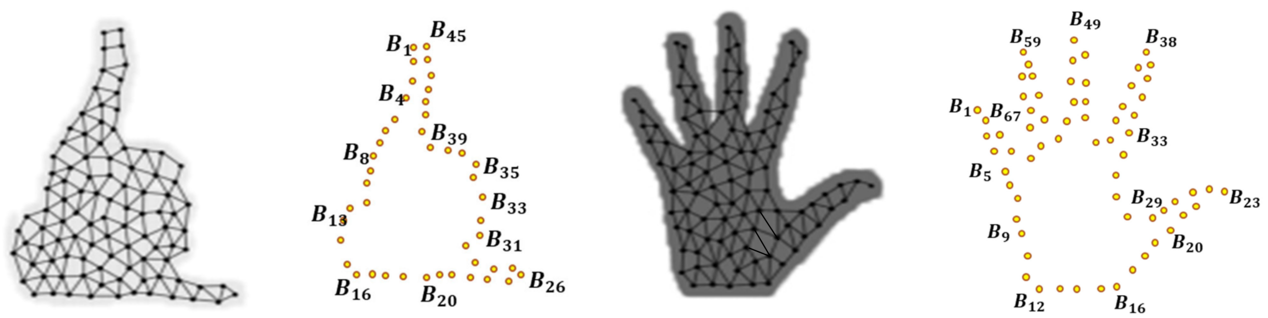
(a) Create a connecting edge if there is no connection between x_1 and x_2.
$$edge = edge\ U\{(x_1, x_2)\}$$
(b) Set the edge=0.
(c) Adjust the error of the winning node x_1:
$$E_{x_1}\ =\ E_{x_1} +\ \parallel\ U -\ w_{x_1}\ \parallel$$
(d) To adjust the winning node x_1 use the learning rate
(e) Adjust all of the edges connected with node x_1:

3. Remove all edges larger than a_{max} and delete all nodes without connecting edges
4. Insert new nodes and divide them into two parts.
5. Insert new nodes in the following steps

i. Locate the neighboring node *n* of u with the largest error, and insert new node r between them.
$$V = V\ U\ \{r\},\ w_r = (\ w_u +\ w_v)/2$$
ii. Create the edges of r with u and v, and delete the edge between u and v and locate the induced subgraph
iii. Lower down the error of u and v, and set the error of node r.
iv. Regulate error of all nodes

6. If stop conditions are not satisfied, then go back to Step 1.

The outer nodes index value is taken as a feature. Each gesture depicts the different morphology of the gesture. The outer nodes are selected by inspecting the neighborhood pixel values. If the pixel has the white value, then the node is selected; otherwise, it is rejected. Figure 11 shows the mesh and the selected outer boundary of the hand.



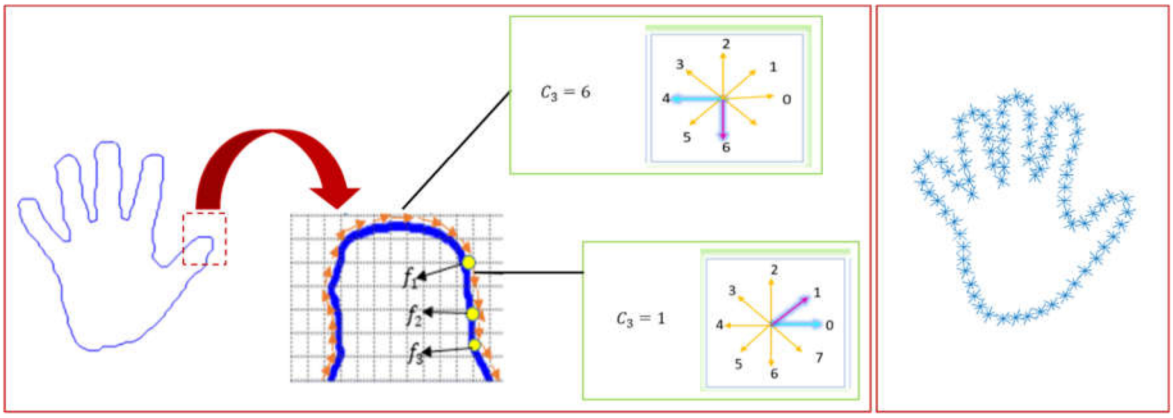
(**a**) (**b**)

Figure 11. (**a**) SONG on the call gesture with the outer region selected as a feature. (**b**) SONG on fine gesture with the outer region selected for feature.

3.5.2. Active Model

The second method used for feature extraction from the full hand uses 8 Freeman chain code algorithms. This method measures the intensity change along with the curve points on the boundary of hand gestures. First, the boundary of the hand is identified. All the curve points along the hand contour are identified and represented using the 8 Freeman chain code algorithm [75,76]. Let us suppose all the points along the boundary *b* are represented by points n. The s is the starting point on the top left side of the thumb, and s will check points until n-1. The curve point on the boundary is represented as C_b, and thus all points will be $C_b = \{s_0, s_1, \ldots \ldots, s_{n-1}\}$.

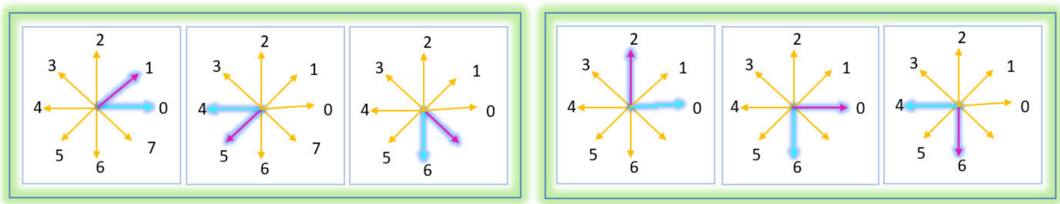
We start to find feature points from s0 and move in a clockwise direction along the boundary until a change in the direction is observed. Let the next point be s1 and current points s0; if the direction of both s0 and s1 is the same, then the point s1 will be excluded and the next point, s2, will be checked. If the directions of both s0 and s1 are different, then s1 will be considered as feature point f. All points on the boundary will be checked similarly and, if the current point and the next point difference is greater than 0, then it will be selected as feature point f [77]. Figure 12 depicts point selection.



(**a**) (**b**)

Figure 12. (**a**) depicts active model feature extraction and the direction of extracted points, while (**b**) depicts the points extracted.

A total of 8 cases of 45° and 4 cases of 90° are taken to find the changes in the direction of points in order to find the points for features. Figure 13 represents the changes in direion of 45° and 90° in which the yellow line shows the direction of the current curve point while the blue arrow shows the subsequent direction of the curve point.



(**a**) (**b**)

Figure 13. Cases of full-hand feature extraction: (**a**) three cases of 45° change in direction; (**b**) three cases of 90° change in direction.

3.6. Features Optimization

For feature optimization, gray wolf optimization (GWO) is applied in order to obtain the best feature vector for classification. GWO discriminates the different cases and provides multiple solutions. It resembles the organizational structure of wolves for group hunting, which is a very clever swarm tactic. Four types of wolves stimulate leadership hierarchy. The alpha wolf is the master for all the gestures. The beta wolf is a subordinate wolf, which also helps the alpha to make choices [51,78,79]. The delta wolf is only appointed when alpha, beta, and omega are not wolves. The omega is a low-rated wolf that only reports to the other wolves. The omega is also dominated by delta wolves and it reports to both alpha and beta. The strategies of hunting that identify a wolf's location can be seen mathematically as

$$\hat{A}\alpha = |\acute{Y}1 \cdot \overline{L}\alpha - \overline{L}(t)|,\ \hat{A}\beta = |\acute{Y}2 \cdot \overline{L}\beta - \overline{L}(t)|, \hat{A}\delta = |\acute{Y}3 \cdot \overline{L}\delta - \overline{L}(t)| \quad (23)$$

where t is iterations. When the target is identified, the repletion begins (t = 1). The alpha, delta, and beta would instruct the omegas to chase and encircle the target. L is the location trajectory of the gray wolf [80]. L is defined as

$$\overline{L}1 = \overline{L}\alpha - \overline{Z}1 \cdot \hat{A}\alpha,\ \overline{L}2 = \overline{L}\beta - \overline{Z}2 \cdot \hat{A}\beta, \overline{L}3 = \overline{L}\delta - \overline{Z}3 \cdot \hat{A}\delta,\ \overline{L}(t) = \frac{\overline{L}1 + \overline{L}2 + \overline{L}3}{3} \quad (24)$$

where $L1$, $L2$, and $L3$ are the location trajectories of alpha, beta, and delta wolves, respectively. The L and d are the mixtures of the containing restriction a and the haphazard quantities $x1$ and $x2$ as

$$\overline{L} = 2\alpha x_1\ ,\ d = 2x_2 \quad (25)$$

The optimization result for the Sign Word dataset is given below (Figure 14):

3.7. Classifier: Reweighted Genetic Algorithm

For classification, a modified version of the state-of-the-art genetic algorithm (GA) is introduced. A genetic algorithm (GA) is an evolutionary algorithm that is robust, heuristic, and stochastic and is reliable for high-dimensional space [81]. The genetic strategy is used during complex computational problems. It is a pool-based algorithm that uses small chunks of data to find optimal solutions with random biological operations, i.e., crossover, mutation, and selection. In the genetic model, operations are performed on a basic unit known as chromosomes. Feature vectors are converted into chromosomes by mapping every single feature to respective genes [82]. Chromosomes consist of genes; each gene represents a single feature in the feature vector. Figure 15 shows the basic structure of genetic model units. To find the optimal solution chromosomes, filter the search space in different orders. On the other hand, the population is the pool of chromosomes. In selection process, the first chromosome is selected randomly from the pool, and after that, a fitness function is applied to all chromosomes and numbers are generated. The chromosome having greater number is the fittest and it is selected for the optimal path solution [83].

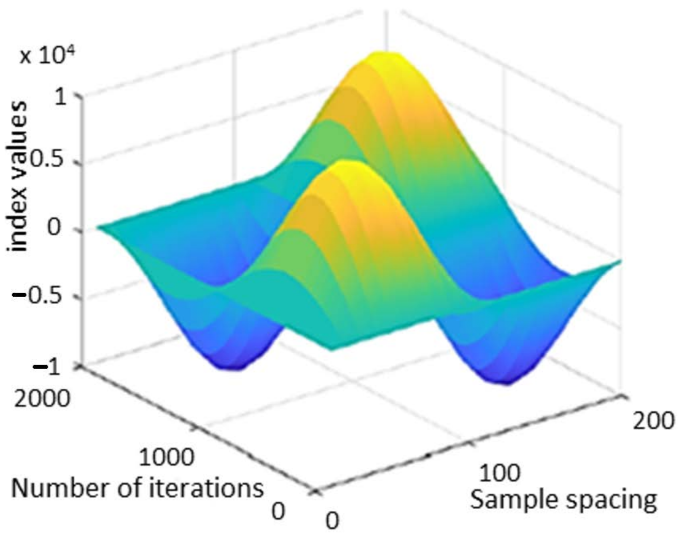


Figure 14. Gray wolf optimization best solution on the Sign Word dataset.

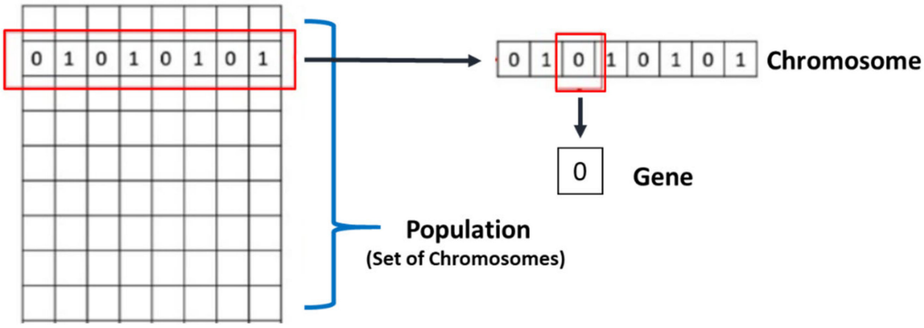


Figure 15. Representation of the basic units of a genetic model.

In the reweighted genetic algorithm, the classifier is divided into 2 phases: reweighted feature selection and classification. In the first phase, weights are assigned to optimized features using a support vector machine and random forest classifier. In the classification phase, the resultant output is classified into different human gestures.

Initially, GA starts with optimized features on which crossover and mutation techniques are applied. In the crossover function, the optimized features are represented as chromosomes in a subspace known as population. After this, mutation is applied to crossed chromosomes to increase diversity. This also provides a method that helps in escaping from the local optimum. Finally, resultant chromosomes are duplicated, and weights are assigned to them so that prominent features are assigned according to better weights.

$$C_{opt}\ (f) = \sum_{k=1}^{K} O_{f1}, O_{f2} \ldots\ldots, O_{fn}, * O_{f1}, O_{f2} \ldots\ldots, O_{fn} M_{opt}(f) = O''_{f1}, O''_{f2} \ldots\ldots, O''_{fn1} \quad (26)$$

where *Of1* is the optimized feature, C_{opt} is the crossover, and M_{opt} is the mutation function applied over gray wolf optimized features. These GA patterns are then inserted into a codebook pattern and classified by finding a maximum matching cluster from the codebook [84] (Figure 16).

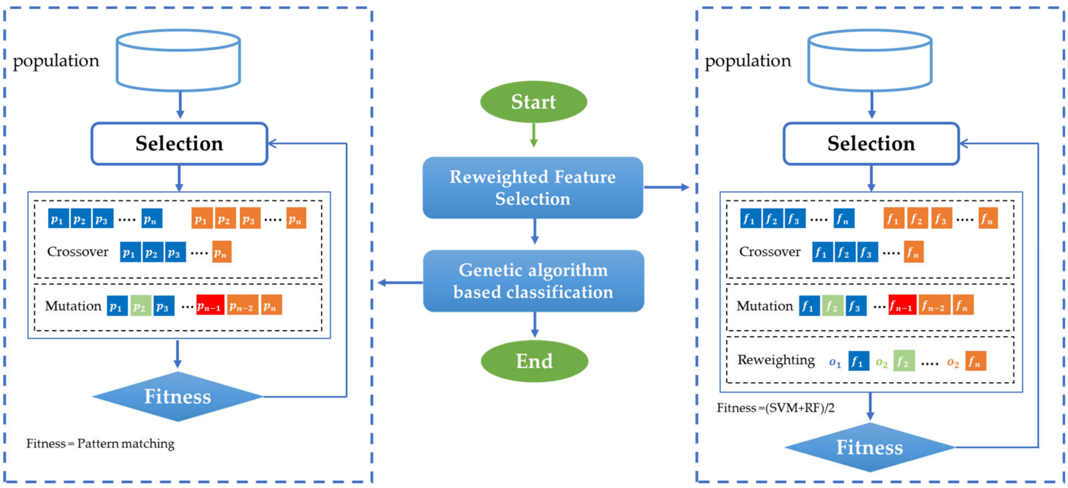


Figure 16. Flow chart of the reweighted genetic model for HGR.

4. System Validation and Experimentation

This section provides a brief description of the datasets used for the training and testing of the proposed system. All the experiments were performed on MATLAB R2017a. The following parameters were used to validate the system's performance. Firstly, the recognition rate of single and gesture performed by both hands from all five datasets is given. Secondly, the precision, recall, and F1 values via decision tree, ANN, and genetic algorithm are given for all five datasets. Finally, a comparison of our method with other state-of-the-art methods is provided.

4.1. Dataset Description

Table 2 represents the name, type of input data, and description of each dataset for the training and testing of the proposed system.

Table 2. Descriptions of datasets used for evaluation and experimentation.

Name of Dataset	Type of Input Data	Gesture Classes
Sign word	RGB images	This dataset contains 20 isolated hand gestures (11 single-hand gestures and 9 double-hand gestures), i.e., call, close, cold, correct, fine, help, home, like, love, no, ok, please, single, sit, tall, wash, work, yes, you, iloveyou. The images of the dataset were collected with a pixel resolution of 200 x 200. To collect dataset images, we requested three volunteers (mean age 25) to perform the gesture of Sign Word [85].
Dexter1	RGB frames with 5 Sony DFW-V500 RGB cameras at 25 fps	Dexter1 consists of seven sequences, i.e., abduction–adduction, flexion–extension, finger count, finger wave, flexex1, pinch, random, tiger grasp of the hand. Roughly the first 250 frames in each sequence correspond to slow motions while the remaining frames are fast motions. All sequences are performed with an actor's right hand [86].
Dexter + Object	RGB frames with Creative Senz3D color camera	Dexter + Object is a dataset for evaluating algorithms for joint hand and object tracking. It consists of six sequences, i.e., grasp1, grasp2, pinch, rigid, rotate, and occlusion with two actors (one female) and varying interactions with a simple object shape [87].
STB	RGB and RGBD frames	STB dataset contains 18,000 images with ground truth. Six people performed counting and random poses having different backgrounds [88].
NYU	RGBD data with ground truth images	NYU hand pose dataset consists of 8252 test set and 72,757 training set frames. The dataset consists of RGBD and RGB images. The training set consists of single user while test set consist of two users with different hand poses [89].

4.2. Recognition Accuracy

To validate the system's performance, we first gave the Sign Word dataset hand gesture to the proposed system to determine the recognition rate using a genetic classifier. The percentage of accuracies for each class was given separately in the form of a confusion matrix. Each gesture class for all five datasets used for experimentation achieved up to the mark performance with our proposed system. Tables 3–5 show the confusion matrix of accuracy scores for gesture classification for the proposed approach for the Sign Word dataset, the Dexter1, and the Dexter + Object, respectively. Table 6 shows the mean accuracy of all five datasets used for testing the proposed system.

We used five HGR datasets for experimentation, namely, Sign Word, Dexter1, Dexter + Object, STB, and NYU datasets that produced 92.1%, 93.1%, 88.2%, 90.8%, and 85.3% mean accuracy, respectively.

4.3. Precision, Recall, and F1 Score

In this Section, precision accuracy, recall, and F1 scores are given using a decision tree, ANN, and genetic algorithms on all five datasets. Results show that the genetic algorithm produced a better performance over all three classifiers. The decision tree omitted sampling features for classification while training and the classification process was faster compared to training. ANN required a maximum number of samples for training and, as the number of training samples was less than 100 million, the accuracy rate for ANN was less compared to the other classifiers. The genetic algorithm gave better results in the proposed system. Tables 7–11 present the test results for precision, recall, and F1 scores for all the three classifiers on all five respective datasets.

Table 3. Confusion matrix of accuracy scores for gesture classification for the Sign Word dataset.

Classes	C [1]	CL [2]	CO [3]	CR [4]	F [5]	H [6]	HM [7]	L [8]	LV [9]	N [10]	O [11]	P [12]	S [13]	ST [14]	T [15]	WA [16]	WO [17]	Y [18]	U [19]	IL [20]
C [1]	0.99	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
CL [2]	0.00	0.96	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.02	0.00	0.00	0.00	0.00	0.00
CO [3]	0.01	0.00	0.87	0.00	0.00	0.01	0.00	0.00	0.00	0.02	0.00	0.00	0.00	0.01	0.00	0.03	0.05	0.00	0.00	0.00
CR [4]	0.00	0.00	0.00	0.97	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.02	0.00	0.00
F [5]	0.00	0.00	0.00	0.00	0.99	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
H [6]	0.03	0.00	0.01	0.00	0.00	0.85	0.01	0.00	0.02	0.00	0.02	0.00	0.00	0.01	0.00	0.02	0.03	0.00	0.00	0.00
HM [7]	0.00	0.00	0.00	0.00	0.00	0.00	0.95	0.00	0.02	0.00	0.00	0.00	0.00	0.00	0.00	0.03	0.00	0.00	0.00	0.00
L [8]	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.96	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.02
LV [9]	0.00	0.00	0.01	0.00	0.00	0.03	0.00	0.00	0.94	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.01	0.00	0.00
N [10]	0.01	0.00	0.00	0.02	0.00	0.00	0.00	0.01	0.00	0.91	0.01	0.00	0.00	0.01	0.00	0.00	0.00	0.01	0.02	0.00
O [11]	0.02	0.00	0.00	0.01	0.01	0.00	0.00	0.03	0.00	0.03	0.86	0.01	0.01	0.00	0.00	0.00	0.00	0.01	0.01	0.00
P [12]	0.00	0.00	0.00	0.00	0.03	0.00	0.00	0.00	0.00	0.00	0.00	0.97	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
S [13]	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.03	0.00	0.96	0.00	0.00	0.00	0.00	0.00	0.00	0.01
ST [14]	0.00	0.00	0.03	0.00	0.00	0.02	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.85	0.00	0.07	0.02	0.00	0.00	0.00
T [15]	0.00	0.04	0.01	0.00	0.00	0.02	0.03	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.89	0.00	0.01	0.00	0.00	0.00
WA [16]	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.02	0.00	0.00	0.00	0.00	0.02	0.00	0.87	0.08	0.00	0.00	0.00
WO [17]	0.00	0.00	0.04	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.05	0.89	0.02	0.00	0.00
Y [18]	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.03	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.88	0.07	0.00
U [19]	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.08	0.89	0.00
IL [20]	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.03	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.97

[1] call, [2] close, [3] cold, [4] correct, [5] fine, [6] help, [7] home, [8] like, [9] love, [10] no, [11] ok, [12] please, [13] single, [14] sit, [15] tall, [16] wash, [17] work, [18] yes, [19] you, [20] iLoveYou.

Table 4. Confusion matrix of accuracy scores for gesture classification for the Dexter1 dataset.

Gesture Classes	AD [1]	FC [2]	FW [3]	F [4]	P [5]	R [6]	TG [7]
AD [1]	0.95	0.0	0.00	0.05	0.00	0.00	0.00
FC [2]	0.03	0.96	0.00	0.00	0.00	0.00	0.02
FW [3]	0.00	0.04	0.94	0.02	0.00	0.00	0.00
F [4]	0.03	0.00	0.02	0.95	0.00	0.00	0.00
P [5]	0.01	0.01	0.00	0.00	0.92	0.00	0.06
R [6]	0.00	0.03	0.05	0.00	0.03	0.89	0.0
TG [7]	0.00	0.00	0.00	0.02	0.07	0.00	0.91

[1] adbadd, [2] fingercount, [3] fingerwave, [4] flexer1, [5] pinch, [6] random, [7] tigergrasp.

Table 5. Confusion matrix of accuracy scores for gesture classification for the Dexter + Object dataset.

	Predicted Gesture Classes					
	G [1]	**GR [2]**	**O [3]**	**P [4]**	**R [5]**	**RO [6]**
G [1]	0.91	0.07	0.00	0.00	0.00	0.02
GR [2]	0.09	0.89	0.00	0.00	0.01	0.01
O [3]	0.00	0.00	0.85	0.00	0.09	0.06
P [4]	0.06	0.05	0.00	0.84	0.03	0.02
R [5]	0.00	0.01	0.06	0.00	0.92	0.01
RO [6]	0.00	0.00	0.00	0.05	0.07	0.88

[1] grasp1, [2] grasp2, [3] occlusion, [4] pinch, [5] rigid, [6] rotate.

Table 6. Mean accuracy for gesture classification of datasets.

Datasets	Mean Accuracy %
Sign Word	92.1
Dexter1	93.1
Dexter + Object	88.2
STB	90.8
NYU	85.3

Table 7. Test results of the three classifiers using the Sign Word dataset.

Classifier	Accuracy	Precision	Recall	F1
Decision tree	0.9142	0.9012	0.8412	0.8701
ANN	0.8924	0.8214	0.8516	0.8362
Genetic algo	0.9212	0.8817	0.8833	0.8824

Table 8. Test results of the three classifiers using the Dexter1 dataset.

Classifier	Accuracy	Precision	Recall	F1
Decision tree	0.9212	0.9102	0.8702	0.8897
ANN	0.9024	0.8313	0.8714	0.8508
Genetic algo	0.9312	0.8927	0.8923	0.8924

Table 9. Test results of the three classifiers using the Dexter + Object dataset.

Classifier	Accuracy	Precision	Recall	F1
Decision tree	0.9021	0.9012	0.8412	0.8701
ANN	0.8761	0.7915	0.8315	0.8110
Genetic algo	0.8822	0.8315	0.8012	0.8160

Table 10. Test results of the three classifiers using the STB dataset.

Classifier	Accuracy	Precision	Recall	F1
Decision tree	0.8901	0.8542	0.8612	0.8576
ANN	0.8912	0.8612	0.8415	0.8512
Genetic algo	0.9081	0.8522	0.8724	0.8621

Table 11. Test results of the three classifiers using the NYU dataset.

Classifier	Accuracy	Precision	Recall	F1
Decision tree	0.8641	0.8414	0.8213	0.8312
ANN	0.8421	0.8321	0.8101	0.8214
Genetic algo	0.8532	0.8462	0.8387	0.8424

4.4. Comparison

The comparison between our proposed method and other state-of-art-methods is given in Table 12. The results show that our proposed method, which is the combined feature extraction method (i.e., using both key points and full hand), produced higher recognition accuracy rates than the other state-of-the-art methods, which use a single feature extraction method (i.e., either point based or full hand). Our proposed method accurately extracted ROI from RGB images and accurately extracted feature vectors on the proposed method. The reweighted genetic algorithm used optimized features, 70% of the feature vectors for training and 30% of the feature vector for testing, to produce accurate results. The table shows that on all five datasets, namely, Sign Word, Dexter1, Dexter + Object, STB, and NYU used for training and testing, our proposed method produced higher accuracy than the other methods.

Table 12. Result comparison with the other state-of-the-art methods on all three datasets.

<table>
<tr><th>Dataset</th><th>Feature Extraction Method</th><th>Authors</th><th>Recognition Accuracy (%)</th></tr>
<tr><td rowspan="4">Sign Word</td><td>Point-based</td><td>Vaitkevičius et al. [90]</td><td>86.1</td></tr>
<tr><td rowspan="2">Full-hand</td><td>Ahlawat et al. [91]</td><td>90</td></tr>
<tr><td>Wang et al. [92]</td><td>92</td></tr>
<tr><td>Point-based + full-hand</td><td>Proposed methodology on Sign Word dataset</td><td>92.1</td></tr>
<tr><td rowspan="4">Dexter1</td><td>Point-based</td><td>Cai [93]</td><td>88</td></tr>
<tr><td rowspan="2">Full-hand</td><td>Imashev [94]</td><td>86</td></tr>
<tr><td>Shan et al. [95]</td><td>89</td></tr>
<tr><td>Point-based + full-hand</td><td>Proposed methodology on Dexter1 dataset</td><td>93.1</td></tr>
<tr><td rowspan="4">Dexter + Object</td><td rowspan="2">Point-based</td><td>Spurr et al. [96]</td><td>85</td></tr>
<tr><td>Brahmbhatt et al. [97]</td><td>86.49</td></tr>
<tr><td>Full-hand</td><td>Li et al. [98]</td><td>84</td></tr>
<tr><td>Point-based + full-hand</td><td>Proposed methodology on Dexter + Object dataset</td><td>88.2</td></tr>
<tr><td rowspan="4">STB</td><td rowspan="2">Point-based</td><td>Chen et al. [99]</td><td>75</td></tr>
<tr><td>Dai et al. [100]</td><td>77</td></tr>
<tr><td>Full-hand</td><td>Zhou et al. [101]</td><td>89</td></tr>
<tr><td>Point-based + full-hand</td><td>Proposed methodology on STB</td><td>90.8</td></tr>
<tr><td rowspan="3">NYU</td><td>Point-based</td><td>Deng et al. [102]</td><td>74</td></tr>
<tr><td>Full-hand</td><td>Moon et al. [103]</td><td>83.4</td></tr>
<tr><td>Point-based + full-hand</td><td>Proposed methodology on NYU</td><td>85.3</td></tr>
</table>

5. Conclusions

In this research work, we developed an efficient HGR system for healthcare muscle exercise via point-based and full-hand features and a reweighted genetic algorithm. Features proposed in this method include Euclidean distance, the cosine of angles (i.e., α, β, γ), area of irregular shapes, SONG mesh, and chain model to select the optimal features. GWO with RGA is used to optimize, train, and recognize different gestures for muscle exercise. Our proposed system outperformed other HGR systems in terms of accuracy at 92.1%, 93.1%, 88.2%, 90.8%, and 85.3% over the Sign Word, Dexter1, Dexter + Object, STB, and NYU datasets, respectively. Then, precision, recall, and F1 scores were also measured for overall gesture recognition in all datasets. In the end, the performance of the proposed system was compared with the other state-of-the-art systems. We expect our system to perform well for the recognition of daily gestures performed in any environment.

In the future, we plan to improve features with different techniques. 3D mesh features will be improved. We will also develop our dataset for healthcare, which will include complex gestures. Dynamic gestures will also be tackled and recognized by the system.

Author Contributions: Conceptualization, H.A.; methodology, H.A., M.G., and A.J.; software, H.A.; validation, M.G. and A.J.; formal analysis, K.K. and M.G.; resources, A.J. and K.K.; writing—review and editing, A.J. and K.K.; funding acquisition, A.J. and K.K. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by the Basic Science Research Program through the National Research Foundation of Korea (NRF), funded by the Ministry of Education (no. 2018R1D1A1A02085645). Moreover, this work was supported by the Korea Medical Device Development Fund grant funded by the Korean government (the Ministry of Science and ICT; the Ministry of Trade, Industry and Energy; the Ministry of Health and Welfare; and the Ministry of Food and Drug Safety) (project number: 202012D05-02).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Liu, H.; Lihui, W. Gesture recognition for human-robot collaboration: A review. *Int. J. Ind. Ergon.* **2018**, *68*, 355–367. [CrossRef]
2. Tingting, Y.; Junqian, W.; Lintai, W.; Yong, X. Three-stage network for age estimation. *CAAI Trans. Intell. Technol.* **2019**, *4*, 122–126. [CrossRef]
3. Haria, A.P.; BMS College of Engineering; Subramanian, A.; Asokkumar, N.; Podda, S.; Nayak, J.S. Hand Gesture Recognition System. *Int. J. Comput. Trends Technol.* **2017**, *47*, 209–212. [CrossRef]
4. Nishihara, H.K.; Hsu, S.P.; Kaehler, A.; Jangaard, L. Northrop Grumman Systems Corp. Hand-Gesture Recognition Method. U.S. Patent No. 9,696,808, April 2017.
5. Sagayam, K.M.; Hemanth, D.J. Hand posture and gesture recognition techniques for virtual reality applications: A survey. *Virtual Real.* **2017**, *21*, 91–107. [CrossRef]
6. Bobic, V.; Tadic, P.; Kvascev, G. Hand gesture recognition using neural network based techniques. In Proceedings of the 2016 13th Symposium on Neural Networks and Applications (NEUREL), Belgrade, Serbia, 22–24 November 2016; pp. 1–4. [CrossRef]
7. Oudah, M.; Al-Naji, A.; Chahl, J. Hand Gesture Recognition Based on Computer Vision: A Review of Techniques. *J. Imaging* **2020**, *6*, 73. [CrossRef]
8. Li, W.-J.; Hsieh, C.-Y.; Lin, L.-F.; Chu, W.-C. Hand gesture recognition for post-stroke rehabilitation using leap motion. In Proceedings of the 2017 International Conference on Applied System Innovation (ICASI), Sapporo, Japan, 13–17 May 2017; IEEE: Piscataway, NJ, USA, 2017; pp. 386–388. [CrossRef]
9. Cheng, J.; Wei, F.; Liu, Y.; Li, C.; Chen, Q.; Chen, X. Chinese Sign Language Recognition Based on DTW-Distance-Mapping Features. *Math. Probl. Eng.* **2020**, *2020*, 1–13. [CrossRef]
10. Jalal, A.; Uddin, I. Security architecture for third generation (3G) using GMHS cellular network. In Proceedings of the 2007 International Conference on Emerging Technologies, Rawalpindi, Pakistan, 12–13 November 2007; pp. 74–79. [CrossRef]
11. Oyedotun, O.K.; Khashman, A. Deep learning in vision-based static hand gesture recognition. *Neural Comput. Appl.* **2017**, *28*, 3941–3951. [CrossRef]
12. Pinto, R.F.; Borges, C.D.B.; Almeida, A.M.A.; Paula, I.C. Static Hand Gesture Recognition Based on Convolutional Neural Networks. *J. Electr. Comput. Eng.* **2019**, *2019*, 1–12. [CrossRef]
13. Gao, Q.; Liu, J.; Ju, Z.; Li, Y.; Zhang, T.; Zhang, L. Static Hand Gesture Recognition with Parallel CNNs for Space Human-Robot Interaction. In *Constructive Side-Channel Analysis and Secure Design, Proceedings of the 8th International Workshop, COSADE 2017, Paris, France, 13–14 April 2017*; Springer International Publishing: Singapore, 2017; pp. 462–473. [CrossRef]

14. Zheng, Q.; Yang, M.; Tian, X.; Jiang, N.; Wang, D. A Full Stage Data Augmentation Method in Deep Convolutional Neural Network for Natural Image Classification. *Discret. Dyn. Nat. Soc.* **2020**, *2020*, 1–11. [CrossRef]
15. Asif, A.R.; Waris, A.; Gilani, S.O.; Jamil, M.; Ashraf, H.; Shafique, M.; Niazi, I.K. Performance Evaluation of Convolutional Neural Network for Hand Gesture Recognition Using EMG. *Sensors* **2020**, *20*, 1642. [CrossRef]
16. Su, H.; Ovur, S.E.; Zhou, X.; Qi, W.; Ferrigno, G.; De Momi, E. Depth vision guided hand gesture recognition using electromyographic signals. *Adv. Robot.* **2020**, *34*, 985–997. [CrossRef]
17. Motoche, C.; Benalcázar, M.E. Real-Time Hand Gesture Recognition Based on Electromyographic Signals and Artificial Neural Networks. In *Constructive Side-Channel Analysis and Secure Design, Proceedings of the 9th International Workshop, COSADE 2018, Singapore, 23–24 April 2018*; Springer International Publishing: Singapore, 2018; pp. 352–361. [CrossRef]
18. Sapienza, S.; Ros, P.M.; Guzman, D.A.F.; Rossi, F.; Terracciano, R.; Cordedda, E.; Demarchi, D. On-Line Event-Driven Hand Gesture Recognition Based on Surface Electromyographic Signals. In Proceedings of the 2018 IEEE International Symposium on Circuits and Systems (ISCAS), Florence, Italy, 27–30 May 2018; pp. 1–5. [CrossRef]
19. Pinzon-Arenas, J.O.; Jimenez-Moreno, R.; Herrera-Benavides, J.E. Convolutional Neural Network for Hand Gesture Recognition using 8 different EMG Signals. In Proceedings of the 2019 XXII Symposium on Image, Signal Processing and Artificial Vision (STSIVA), Bucaramanga, Colombia, 24–26 April 2019; pp. 1–5. [CrossRef]
20. Benalcazar, M.E.; Jaramillo, A.G.; Jonathan; Zea, A.; Paez, A.; Andaluz, V.H. Hand gesture recognition using machine learning and the Myo armband. In *Proceedings of the 2017 25th European Signal Processing Conference (EUSIPCO), Kos Island, Greece, 28 August–2 September 2017*; IEEE: Piscataway, NJ, USA, 2017; pp. 1040–1044. [CrossRef]
21. Qi, J.; Jiang, G.; Li, G.; Sun, Y.; Tao, B. Surface EMG hand gesture recognition system based on PCA and GRNN. *Neural Comput. Appl.* **2019**, *32*, 6343–6351. [CrossRef]
22. Qi, W.; Su, H.; Aliverti, A. A Smartphone-Based Adaptive Recognition and Real-Time Monitoring System for Human Activities. *IEEE Trans. Hum. Mach. Syst.* **2020**, *50*, 414–423. [CrossRef]
23. Wang, Z.; Hou, Y.; Jiang, K.; Dou, W.; Zhang, C.; Huang, Z.; Guo, Y. Hand Gesture Recognition Based on Active Ultrasonic Sensing of Smartphone: A Survey. *IEEE Access* **2019**, *7*, 111897–111922. [CrossRef]
24. Haseeb, M.A.A.; Parasuraman, R. Wisture: Touch-Less Hand Gesture Classification in Unmodified Smartphones Using Wi-Fi Signals. *IEEE Sens. J.* **2018**, *19*, 257–267. [CrossRef]
25. Zhang, H.; Xu, W.; Chen, C.; Bai, L.; Zhang, Y. Your Knock Is My Command: Binary Hand Gesture Recognition on Smartphone with Accelerometer. *Mob. Inf. Syst.* **2020**, *2020*, 1–16. [CrossRef]
26. Panella, M.; Altilio, R. A Smartphone-Based Application Using Machine Learning for Gesture Recognition: Using Feature Extraction and Template Matching via Hu Image Moments to Recognize Gestures. *IEEE Consum. Electron. Mag.* **2018**, *8*, 25–29. [CrossRef]
27. Aldabbagh, G.; AlGhazzawi, D.M.; Hasan, S.H.; Alhaddad, M.; Malibari, A.; Cheng, L. Optimal Learning Behavior Prediction System Based on Cognitive Style Using Adaptive Optimization-Based Neural Network. *Complexity* **2020**, *2020*, 1–13. [CrossRef]
28. Wiens, T. Engine speed reduction for hydraulic machinery using predictive algorithms. *Int. J. Hydromech.* **2019**, *2*, 16. [CrossRef]
29. Li, G.; Tang, H.; Sun, Y.; Kong, J.; Jiang, G.; Jiang, D.; Tao, B.; Xu, S.; Liu, H. Hand gesture recognition based on convolution neural network. *Clust. Comput.* **2019**, *22*, 2719–2729. [CrossRef]
30. Zheng, Q.; Tian, X.; Liu, S.; Yang, M.; Wang, H.; Yang, J. Static Hand Gesture Recognition Based on Gaussian Mixture Model and Partial Differential Equation. *IAENG Int. J. Comput.* **2018**, *Sci. 45*, 569–583.
31. Cheng, H.; Dai, Z.; Liu, Z.; Zhao, Y. An image-to-class dynamic time warping approach for both 3D static and trajectory hand gesture recognition. *Pattern Recognit.* **2016**, *55*, 137–147. [CrossRef]
32. Oprisescu, S.; Christoph, R.; Bochao, S. Automatic static hand gesture recognition using tof cameras. In Proceedings of the 20th European Signal Processing Conference (EUSIPCO), Bucharest, Romania, 27–31 August 2012; IEEE: Piscataway, NJ, USA, 2012; pp. 2748–2751.
33. Yun, L.; Lifeng, Z.; Shujun, Z. A Hand Gesture Recognition Method Based on Multi-Feature Fusion and Template Matching. *Proc. Eng.* **2012**, *29*, 1678–1684. [CrossRef]
34. Ghosh, D.K.; Ari, S. Static Hand Gesture Recognition Using Mixture of Features and SVM Classifier. In Proceedings of the 2015 Fifth International Conference on Communication Systems and Network Technologies, Gwalior, India, 4–6 April 2015; IEEE: Piscataway, NJ, USA, 2015; pp. 1094–1099. [CrossRef]
35. Candrasari Banuwati, E.; Novamizanti, L.; Aulia, S. Discrete Wavelet Transform on static hand gesture recognition. *J. Phys. Conf. Ser.* **2019**, *1367*, 012022. [CrossRef]
36. Jalal, A.; Khalid, N.; Kim, K. Automatic Recognition of Human Interaction via Hybrid Descriptors and Maximum Entropy Markov Model Using Depth Sensors. *Entropy* **2020**, *22*, 817. [CrossRef] [PubMed]
37. Chen, X.; Shi, C.; Liu, B. Static hand gesture recognition based on finger root-center-angle and length weighted Ma-halanobis distance. In *Real-Time Image and Video Processing*; International Society for Optics and Photonics: Bellingham, WA, USA, 2016; Volume 9897, p. 98970U. [CrossRef]
38. Bhavana, V.; Mouli, G.M.S.; Lokesh, G.V.L. Hand Gesture Recognition Using Otsu's Method. In Proceedings of the 2017 IEEE International Conference on Computational Intelligence and Computing Research (ICCIC), Tamilnadu, India, 14–16 December 2017; IEEE: Piscataway, NJ, USA, 2017; pp. 1–4. [CrossRef]

39. Yusnita, L.; Rosalina, R.; Roestam, R.; Wahyu, R.B. Implementation of Real-Time Static Hand Gesture Recognition Using Artificial Neural Network. *Commun. Inf. Technol. J.* **2017**, *11*, 85–91. [CrossRef]
40. Jalal, A.; Quaid, M.A.K.; Tahir, S.B.U.D.; Kim, K. A Study of Accelerometer and Gyroscope Measurements in Physical Life-Log Activities Detection Systems. *Sensors* **2020**, *20*, 6670. [CrossRef] [PubMed]
41. Liu, K.; Kehtarnavaz, N. Real-time robust vision-based hand gesture recognition using stereo images. *J. Real Time Image Process.* **2013**, *11*, 201–209. [CrossRef]
42. Ahmed, W.; Chanda, K.; Mitra, S. Vision based Hand Gesture Recognition using Dynamic Time Warping for Indian Sign Language. In Proceedings of the 2016 International Conference on Information Science (ICIS), Kochi, India, 11–14 December 2016; IEEE: Piscataway, NJ, USA, 2017; pp. 120–125. [CrossRef]
43. Al-Shamayleh, A.S.; Ahmad, R.; Abushariah, M.A.M.; Alam, K.A.; Jomhari, N. A systematic literature review on vision based gesture recognition techniques. *Multim. Tools Appl.* **2018**, *77*, 28121–28184. [CrossRef]
44. Pansare, J.R.; Ingle, M. Vision-based approach for American Sign Language recognition using Edge Orientation Histogram. In Proceedings of the 2016 International Conference on Image, Vision and Computing (ICIVC), Portsmouth, NH, USA, 3–5 August 2016; IEEE: Piscataway, NJ, USA, 2016; pp. 86–90. [CrossRef]
45. Hussain, S.; Saxena, R.; Han, X.; Khan, J.A.; Shin, H. Hand gesture recognition using deep learning. In Proceedings of the 2017 International SoC Design Conference (ISOCC), Seoul, Korea, 5–8 November 2017; IEEE: Piscataway, NJ, USA, 2017; pp. 48–49. [CrossRef]
46. Mahdikhanlou, K.; Ebrahimnezhad, H. Multimodal 3D American sign language recognition for static alphabet and numbers using hand joints and shape coding. *Multim. Tools Appl.* **2020**, *79*, 22235–22259. [CrossRef]
47. Liu, J.; Ding, H.; Shahroudy, A.; Duan, L.-Y.; Jiang, X.; Wang, G.; Kot, A.C. Feature Boosting Network For 3D Pose Estimation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2020**, *42*, 494–501. [CrossRef]
48. Jalal, A.; Akhtar, I.; Kim, K. Human Posture Estimation and Sustainable Events Classification via Pseudo-2D Stick Model and K-ary Tree Hashing. *Sustainability* **2020**, *12*, 9814. [CrossRef]
49. Kerdvibulvech, C. A methodology for hand and finger motion analysis using adaptive probabilistic models. *EURASIP J. Embed. Syst.* **2014**, *2014*, 18. [CrossRef]
50. Nguyen, T.-N.; Vo, D.-H.; Huynh, H.-H.; Meunier, J. Geometry-based static hand gesture recognition using support vector machine. In Proceedings of the 2014 13th International Conference on Control Automation Robotics & Vision (ICARCV), Singapore, 10–12 December 2014; IEEE: Piscataway, NJ, USA, 2014; pp. 769–774. [CrossRef]
51. Jalal, A.; Kim, Y.-H.; Kim, Y.-J.; Kamal, S.; Kim, D. Robust human activity recognition from depth video using spatiotemporal multi-fused features. *Pattern Recognit.* **2017**, *61*, 295–308. [CrossRef]
52. Jalal, A.; Uddin, Z.; Kim, T.-S. Depth video-based human activity recognition system using translation and scaling invariant features for life logging at smart home. *IEEE Trans. Consum. Electron.* **2012**, *58*, 863–871. [CrossRef]
53. Kolkur, S.; Kalbande, D.; Shimpi, P.; Bapat, C.; Jatakia, J. Human Skin Detection Using RGB, HSV and YCbCr Color Models. In Proceedings of the International Conference on Communication and Signal Processing 2016 (ICCASP 2016), Lonere, India, 26–27 December 2016; Atlantic Press: Amsterdam, The Netherlands, 2017. [CrossRef]
54. Lv, Y.; Zhou, W. Hierarchical Multimodal Adaptive Fusion (HMAF) Network for Prediction of RGB-D Saliency. *Comput. Intell. Neurosci.* **2020**, *2020*, 1–9. [CrossRef]
55. Zhang, Q.; Yang, M.; Kpalma, K.; Zheng, Q.; Zhang, X. Segmentation of hand posture against complex backgrounds based on saliency and skin colour detection. *IAENG Int. J. Comput. Sci.* **2018**, *45*, 435–444.
56. Grzejszczak, T.; Kawulok, M.; Galuszka, A. Hand landmarks detection and localization in color images. *Multim. Tools Appl.* **2016**, *75*, 16363–16387. [CrossRef]
57. Jalal, A.; Batool, M.; Kim, K. Sustainable Wearable System: Human Behavior Modeling for Life-Logging Activities Using K-Ary Tree Hashing Classifier. *Sustainability* **2020**, *12*, 10324. [CrossRef]
58. Kim, T.; Jalal, A.; Han, H.; Jeon, H.; Kim, J. Real-Time Life Logging via Depth Imaging-based Human Activity Recognition towards Smart Homes Services. In Proceedings of the International Symposium on Renewable Energy Sources and Healthy Buildings, Seoul, Korea, 26–29 August 2014; p. 63. [CrossRef]
59. Tahir, S.B.U.D.; Jalal, A.; Kim, K. Wearable Inertial Sensors for Daily Activity Analysis Based on Adam Optimization and the Maximum Entropy Markov Model. *Entropy* **2020**, *22*, 579. [CrossRef]
60. Jalal, A.; Batool, M.; Kim, K. Stochastic Recognition of Physical Activity and Healthcare Using Tri-Axial Inertial Wearable Sensors. *Appl. Sci.* **2020**, *10*, 7122. [CrossRef]
61. Ahmed, A.; Jalal, A.; Kim, K. A Novel Statistical Method for Scene Classification Based on Multi-Object Categorization and Logistic Regression. *Sensors* **2020**, *20*, 3871. [CrossRef] [PubMed]
62. Mahmood, M.; Jalal, A.; Kim, K. WHITE STAG model: Wise human interaction tracking and estimation (WHITE) using spa-tio-temporal and angular-geometric (STAG) descriptors. *Multim. Tools Appl.* **2019**, *79*, 1–32. [CrossRef]
63. Shehzed, A.; Jalal, A.; Kim, K. Multi-Person Tracking in Smart Surveillance System for Crowd Counting and Normal/Abnormal Events Detection. In Proceedings of the 2019 International Conference on Applied and Engineering Mathematics (ICAEM), London, UK, 3–5 July 2019; Volume 12, pp. 163–168. [CrossRef]

64. Jalal, A.; Quaid, M.A.K.; Hasan, A.S. Wearable Sensor-Based Human Behavior Understanding and Recognition in Daily Life for Smart Environments. In Proceedings of the 2018 International Conference on Frontiers of Information Technology (FIT), Islamabad, Pakistan, 17–19 December 2018; IEEE: Piscataway, NJ, USA, 2018; pp. 105–110. [CrossRef]
65. Jalal, A.; Sharif, N.; Kim, J.T.; Kim, T.-S. Human activity recognition via recognized body parts of human depth silhouettes for residents monitoring services at smart homes. *Indoor Built Environ.* **2013**, *22*, 271–279. [CrossRef]
66. Jalal, A.; Kamal, S.; Kim, D. A Depth Video Sensor-Based Life-Logging Human Activity Recognition System for Elderly Care in Smart Indoor Environments. *Sensors* **2014**, *14*, 11735–11759. [CrossRef]
67. Susan, S.; Agrawal, P.; Mittal, M.; Bansal, S. New shape descriptor in the context of edge continuity. *CAAI Trans. Intell. Technol.* **2019**, *4*, 101–109. [CrossRef]
68. Osterland, S.; Weber, J. Analytical analysis of single-stage pressure relief valves. *Int. J. Hydromechatron.* **2019**, *2*, 32–53. [CrossRef]
69. Ghesmoune, M.; Lebbah, M.; Azzag, H. A new Growing Neural Gas for clustering data streams. *Neural Netw.* **2016**, *78*, 36–50. [CrossRef]
70. Sun, Q.; Liu, H.; Harada, T. Online growing neural gas for anomaly detection in changing surveillance scenes. *Pattern Recognit.* **2017**, *64*, 187–201. [CrossRef]
71. Zhong, C.; Zhang, B.; Wang, J. Scale-Adaptive Growing Neural Network Based on Distortion Error Stability and its Application in Image Topological Feature Extraction. *IEEE Access* **2021**, *9*, 767–776. [CrossRef]
72. Ghaderi, A.; Morovati, V.; Dargazany, R. A Physics-Informed Assembly of Feed-Forward Neural Network Engines to Predict Inelasticity in Cross-Linked Polymers. *Polymers* **2020**, *12*, 2628. [CrossRef] [PubMed]
73. Jedynak, R. Approximation of the inverse Langevin function revisited. *Rheol. Acta* **2014**, *54*, 29–39. [CrossRef]
74. Hossain, M.; Steinmann, P. More hyperelastic models for rubber-like materials: Consistent tangent operators and comparative study. *J. Mech. Behav. Mater.* **2013**, *22*, 27–50. [CrossRef]
75. Kroon, M. An 8-chain Model for Rubber-like Materials Accounting for Non-affine Chain Deformations and Topological Constraints. *J. Elast.* **2010**, *102*, 99–116. [CrossRef]
76. Hong, F.; Lu, C.; Liu, C.; Liu, R.; Jiang, W.; Ju, W.; Wang, T. PGNet: Pipeline Guidance for Human Key-Point Detection. *Entropy* **2020**, *22*, 369. [CrossRef]
77. Pławiak, P. Novel methodology of cardiac health recognition based on ECG signals and evolutionary-neural system. *Expert Syst. Appl.* **2018**, *92*, 334–349. [CrossRef]
78. Demidova, L.; Nikulchev, E.; Sokolova, Y. The SVM Classifier Based on the Modified Particle Swarm Optimization. *Int. J. Adv. Comput. Sci. Appl.* **2016**, *7*. [CrossRef]
79. Emary, E.; Zawbaa, H.M.; Grosan, C. Experienced Gray Wolf Optimization through Reinforcement Learning and Neural Networks. *IEEE Trans. Neural Netw. Learn. Syst.* **2017**, *29*, 681–694. [CrossRef]
80. Lessmann, S.; Stahlbock, R.; Crone, S. Genetic Algorithms for Support Vector Machine Model Selection. In Proceedings of the 2006 IEEE International Joint Conference on Neural Network Proceedings, Vancouver, BC, Canada, 16–21 July 2006; IEEE: Piscataway, NJ, USA, 2006; pp. 3063–3069. [CrossRef]
81. Quaid, M.A.K.; Jalal, A. Wearable sensors based human behavioral pattern recognition using statistical features and re-weighted genetic algorithm. *Multim. Tools Appl.* **2020**, *79*, 6061–6083. [CrossRef]
82. Batool, M.; Jalal, A.; Kim, K. Telemonitoring of Daily Activity Using Accelerometer and Gyroscope in Smart Home Environ-ments. *J. Electr. Eng. Technol.* **2020**, *15*, 2801–2809. [CrossRef]
83. Ong, Y.S.; Nair, P.B.; Keane, A.J. Evolutionary Optimization of Computationally Expensive Problems via Surrogate Modeling. *AIAA J.* **2003**, *41*, 687–696. [CrossRef]
84. Jalal, A.; Lee, S.; Kim, J.T.; Kim, T.-S. Human Activity Recognition via the Features of Labeled Depth Body Parts. In *Computer Vision*; Springer International Publishing: New York, NY, USA, 2012; pp. 246–249. [CrossRef]
85. Rahim, A.; Islam, R.; Shin, J. Non-Touch Sign Word Recognition Based on Dynamic Hand Gesture Using Hybrid Segmentation and CNN Feature Fusion. *Appl. Sci.* **2019**, *9*, 3790. [CrossRef]
86. Sridhar, S.; Oulasvirta, A.; Theobalt, C. Interactive Markerless Articulated Hand Motion Tracking Using RGB and Depth Data. In Proceedings of the 2013 IEEE International Conference on Computer Vision, Sydney, Australia, 1–8 December 2013; IEEE: Piscataway, NJ, USA, 2013; pp. 2456–2463. [CrossRef]
87. Sridhar, S.; Mueller, F.; Zollhöfer, M.; Casas, D.; Oulasvirta, A.; Theobalt, C. Real-Time Joint Tracking of a Hand Manipulating an Object from RGB-D Input. In Proceedings of the European Conference on Computer Vision (ECCV), Amsterdam, The Netherlands, 8–16 October 2016. [CrossRef]
88. Zhang, J.; Jiao, J.; Chen, M.; Qu, L.; Xu, X.; Yang, Q. 3d hand pose tracking and estimation using stereo matching. *arXiv* **2016**, arXiv:2016.1610.07214.
89. Tompson, J.; Stein, M.; LeCun, Y.; Perlin, K. Real-Time Continuous Pose Recovery of Human Hands Using Convolutional Networks. *ACM Trans. Graph.* **2014**, *33*, 1–10. [CrossRef]
90. Vaitkevičius, A.; Taroza, M.; Blažauskas, T.; Damaševičius, R.; Maskeliūnas, R.; Woźniak, M. Recognition of American Sign Language Gestures in a Virtual Reality Using Leap Motion. *Appl. Sci.* **2019**, *9*, 445. [CrossRef]
91. Ahlawat, S.; Batra, V.; Banerjee, S.; Saha, J.; Garg, A.K. Hand Gesture Recognition Using Convolutional Neural Network. In Proceedings of the International Conference on Innovative Computing and Communications. Lecture Notes in Networks and Systems, Delhi, India, 5–6 May 2018; Springer International Publishing: Singapore, 2018; pp. 179–186. [CrossRef]

92. Wang, J.; Liu, T.; Wang, X. Human hand gesture recognition with convolutional neural networks for K-12 double-teachers instruction mode classroom. *Infrared Phys. Technol.* **2020**, *111*, 103464. [CrossRef]
93. Cai, Y.; Ge, L.; Cai, J.; Yuan, J. Weakly-Supervised 3D Hand Pose Estimation from Monocular RGB Images. In *Constructive Side-Channel Analysis and Secure Design, Proceedings of the 9th International Workshop, COSADE 2018, Singapore, 23–24 April 2018*; Springer International Publishing: Singapore, 2018; pp. 678–694. [CrossRef]
94. Imashev, A.; Mukushev, M.; Kimmelman, V.; Sandygulova, A. A Dataset for Linguistic Understanding, Visual Evaluation, and Recognition of Sign Languages: The K-RSL. In Proceedings of the 24th Conference on Computational Natural Language Learning, online, 19–20 November 2020; Association for Computational Linguistics: Stroudsburg, PA, USA, 2020; pp. 631–640. [CrossRef]
95. Shan, D.; Geng, J.; Shu, M.; Fouhey, D.F. Understanding Human Hands in Contact at Internet Scale. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020; IEEE: Piscataway, NJ, USA; pp. 9866–9875. [CrossRef]
96. Spurr, A.; Song, J.; Park, S.; Hilliges, O. *Cross-Modal Deep Variational Hand Pose Estimation*; Institute of Electrical and Electronics Engineers (IEEE): Piscataway, NJ, USA, 2018; pp. 89–98. [CrossRef]
97. Brahmbhatt, S.; Tang, C.; Twigg, C.D.; Kemp, C.C.; Hays, J. ContactPose: A Dataset of Grasps wi.th Object Contact and Hand Pose. In *Constructive Side-Channel Analysis and Secure Design, Proceedings of the 11th International Workshop, COSADE 2020, Lugano, Switzerland, 1–3, April 2020*; Springer International Publishing: Singapore, 2020; pp. 361–378.
98. Li, M.; Gao, Y.; Sang, N. Exploiting Learnable Joint Groups for Hand Pose Estimation. *arXiv* **2021**, arXiv:2021.2012.09496.
99. Chen, L.; Lin, S.Y.; Xie, Y.; Tang, H.; Xue, Y.; Lin, Y.Y.; Xie, X.; Fan, W. Tagan: Tonality-alignment generative adversarial networks for realistic hand pose synthesis. In Proceedings of the 30th British Machine Vision Conference, BMVC, Cardiff, UK, 9–12 September 2019.
100. Dai, S.; Liu, W.; Yang, W.; Fan, L.; Zhang, J. Cascaded Hierarchical CNN for RGB-Based 3D Hand Pose Estimation. *Math. Probl. Eng.* **2020**, *2020*, 1–13. [CrossRef]
101. Zhou, Y.; Habermann, M.; Xu, W.; Habibie, I.; Theobalt, C.; Xu, F. Monocular real-time hand shape and motion capture us-ing multi-modal data. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 16–18 June 2020; pp. 5346–5355.
102. Deng, X.; Yang, S.; Zhang, Y.; Tan, P.; Chang, L.; Wang, H. Hand3d: Hand pose estimation using 3d neural network. *arXiv* **2017**, arXiv:2017.1704.02224.
103. Moon, G.; Chang, J.Y.; Lee, K.M. V2v-posenet: Voxel-to-voxel prediction network for accurate 3d hand and human pose estimation from a single depth map. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 5079–5088.